CLAIMS

1. A method comprising:

representing hardware and software resources of a distributed computer system as model components; and

forming, from the model components, a logical scale-independent model of an application to be implemented by the distributed computer system.

2. A method as recited in claim 1, wherein each model component represents one or more similar resources.

3. A method as recited in claim 1, wherein each model component is depicted in a graphical user interface as a graphical icon.

4. A method as recited in claim 1, wherein the model components have an associated schema that specifies the hardware and software resources represented by the model components.

5. A method as recited in claim 1, wherein the model components comprise a module that is representative of a behavior of the application that is implemented using the hardware and software resources.

6. A method as recited in claim 1, wherein the model components comprise a store that is representative of persistent data storage.

7. A method as recited in claim 1, wherein the model components comprise a port that is representative of a communication access point for the model components.

8. A method as recited in claim 1, wherein the model components comprise a wire that is representative of an allowable communication connection between model components.

9. A method as recited in claim 1, wherein the model components are selected from a group comprising:

a module that is representative of a behavior of the application that is implemented using the hardware and software resources;

a port that is representative of a service access point for the module or the store; and

a wire that is representative of an allowable communication connection between two or more ports.

10. A method as recited in claim 9, wherein the group of the model components further comprises at least one of:

a store that is representative of persistent data storage;

an event source that is representative of a logical connection point for the module or the store from which event messages originate;

an event sink that is representative of a logical connection point for the module or the store to receive the event messages; and an event wire that is representative of an interconnection between the event source and the event sink.

11. A method as recited in claim 1, further comprising storing the scale-independent model on a computer-readable medium.

12. A method as recited in claim 1, further comprising converting the scale-independent model into a blueprint of the server data center, the blueprint specifying the hardware and software resources used to physically implement the application.

13. A computer-readable medium storing computer-executable instructions that, when executed on a computer, perform the method of claim 1.

14. A method comprising:

defining individual model components as abstract functional operations that are physically implemented by one or more computers and one or more software programs executing on the computers, the model components having an associated schema dictating how the functional operations are specified;

interconnecting the model components to logically connect the functional operations; and

generating a scale-independent application from the interconnected model components and the associated schema.

15. A method as recited in claim 14, wherein each model component is depicted in a graphical user interface as a graphical icon.

16. A method as recited in claim 14, wherein the defining comprises entering, via a user interface, a description of elements needed to implement the functional operations.

17. A method as recited in claim 14, wherein each graphical resource represents a set of resources provided by the computers and the software programs, the resources being scalable from one to many.

18. A method as recited in claim 14, wherein the model components are selected from a group comprising:

a module that is representative of a behavior of the application;

a port that is representative of a communication access point for the module; and

a wire that is representative of an interconnection between two or more ports.

19. A method as recited in claim 18, wherein the group of the model components further comprises:

a store that is representative of persistent data storage;

an event source that is representative of a logical connection point for the module or the store from which event messages originate;

an event sink that is representative of a logical connection point for the module or the store to receive the event messages; and

an event wire that is representative of an interconnection between the event source and the event sink.

20. A method as recited in claim 14, further comprising storing the application on a computer-readable medium.

21. A method as recited in claim 14, further comprising converting the scale-independent application into a blueprint that specifies the computers and the software programs used to physically implement the application.

22. A computer-readable medium storing computer-executable instructions that, when executed on a computer, perform the method of claim 14.

23. A method comprising:

representing hardware and software resources of a distributed computer system as model components; and

associating the model components with a schema dictating how the hardware and software resources are specified.

24. A method as recited in claim 23, wherein the model components are selected from a group comprising:

a module that is representative of a behavior that is implemented using the hardware and software resources;

a port that is representative of a communication access point for the module and the store; and

a wire that is representative of an interconnection between two or more ports.

25. A method as recited in claim 24, wherein the group of the model components further comprises:

a store that is representative of persistent data storage;

an event source that is representative of a logical connection point for the module or the store from which event messages originate;

an event sink that is representative of a logical connection point for the module or the store to receive the event messages; and

an event wire that is representative of an interconnection between the event source and the event sink.

26. A method as recited in claim 23, further comprising creating a scale-independent application from the model components and the associated schema.

27. A method as recited in claim 26, further comprising converting the scale-independent application into a blueprint that specifies the hardware and software resources used to physically implement the application on the distributed computer system.

28. A modeling system, comprising:

a set of model components that represent hardware and software resources of a distributed computer system;

a schema associated with the model components that dictate how the resources are specified; and

a user interface to enable a developer to create an application by selecting and interconnecting the model components and specifying the functionality of the model components in accordance with the schema.

29. A modeling system as recited in claim 28, further comprising a converter to convert the application to a blueprint that specifies the hardware and software resources used to physically implement the application on the distributed computer system.

30. A computer-readable medium comprising computer-executable instructions that, when executed on one or more processors, direct a computing device to:

represent hardware and software resources of a distributed computer system as model components;

associate the model components with a schema dictating how the hardware and software resources are specified; and

create an application by specifying the functionality of the model components in accordance with the schema and interconnecting the model components.

31. A computer-readable medium as recited in claim 30, further comprising computer-executable instructions that, when executed on one or more processors, direct a computing device to convert the application to a blueprint that specifies the hardware and software resources used to physically implement the application on the distributed computer system.

32. A system comprising:

means for representing hardware and software resources of a distributed computer system;

means for specifying how the resources represented by the model components are specified; and

means for selecting and interconnecting the model components and specifying the functionality of the model components to create an application.